CELEPAR
INFORMÁTICA
do PARANÁ

PLATAFORMA DE DESENVOLVIMENTO PINHÃO PARANÁ

MANUAL DE UTILIZAÇÃO DO CVS NO ECLIPSE

Agosto – 2007

Sumário de Informações do Documento

Tipo do Documento: Manual		
Título do Documento: MANUAL DE UTILIZAÇÃO DO CVS NO ECLIPSE		
Estado do Documento: Elaborado		
Responsáveis: Natasha Krassuski Fortes, Leslie Harlley Watter		
Palavras-Chaves: cvs, Eclipse		
Resumo:		
Número de páginas: 28		
Software utilizados:		
Versão	**Data**	**Mudanças**
1.0	07/06/2006	
2.0	23/08/2007	Inclusão de Teoria e Passo a passo de criação e gerenciamento de branches

SUMÁRIO

1 TEORIA

1.1 O que é o CVS?

O **CVS**, ou ***Concurrent Version System*** (Sistema de Versões Concorrentes), é um sistema de controle de versão que permite que se trabalhe com diversas versões de arquivos organizados em um diretório e localizados local ou remotamente, mantendo-se suas versões antigas e os logs (registros) de quem e quando manipulou os arquivos.

É especialmente útil para se controlar versões de um software durante seu desenvolvimento, ou para composição colaborativa de um documento.[1]

1.2 Funcionamento do CVS

O CVS utiliza uma arquitetura cliente-servidor: onde um servidor armazena a versão atual do projeto e seu histórico, e os clientes se conectam a esse servidor para obter uma cópia completa do projeto, trabalhar nessa cópia e então devolver suas modificações.

Tipicamente, cliente e servidor devem estar conectados por uma rede local de computadores, ou pela Internet, mas o cliente e o servidor podem estar na mesma máquina se a configuração do CVS for feita de maneira a dar acesso a versões e histórico do projeto, apenas a usuários locais. O servidor geralmente executa em um sistema ao estilo Unix/Linux, enquanto o cliente CVS pode rodar qualquer sistema operacional.

Vários clientes podem editar cópias do mesmo projeto de maneira concorrente. Quando eles confirmam suas alterações, o servidor tenta fazer uma fusão (merge) delas. Se isso não for possível, por exemplo porque mais de um cliente tentou executar alterações na mesma linha do documento, o servidor apenas executa a primeira alteração e informa ao responsável pela segunda alteração que houve conflito, e que será necessário uma intervenção humana. Se a validação, alteração for bem sucedida, o número de versão de cada cliente arquivo envolvido é incrementado, e o servidor CVS escreve uma linha de observação (fornecida pelo usuário),com a data e o autor das alterações em seus arquivos de log.

Clientes podem comparar diferentes versões de um arquivo, pedir um histórico completo das

1 Texto adaptado de http://pt.wikipedia.org/wiki/CVS

alterações, ou baixar uma determinada versão do projeto, ou de uma data específica, não necessariamente a versão mais atual.

Clientes também podem usar o comando "*update*" para manter suas cópias locais atualizadas com a última versão do servidor. Isso elimina a necessidade de se fazer diversos downloads de todo o projeto.

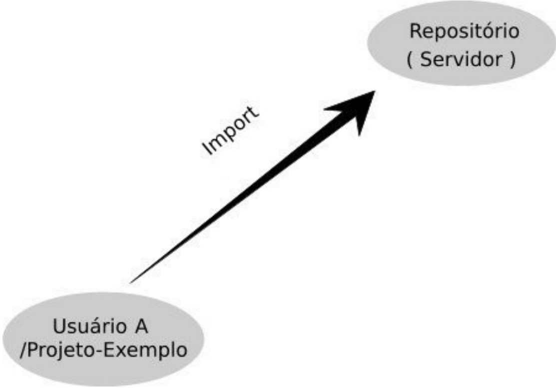
O CVS também pode manter diferentes estados do projeto. Por exemplo, uma versão do software pode ser um desses estados, usado para correção de bugs, enquanto outra versão, que está realmente sob desenvolvimento, sofrendo alterações e tendo novas funcionalidades implementadas, forma o outro estado.

1.3 Terminologia

A terminologia do CVS considera um projeto (conjunto de arquivos relacionados) gerenciados pelo CVS como um **módulo**, que consiste em uma hierarquia de diretórios contendo os arquivos do projeto. Um servidor CVS pode gerenciar diversos módulos; ele armazena todos os módulos administrados por ele em seu **repositório**. A cópia do módulo que foi baixada para um cliente é chamada *cópia de trabalho* (ou *checkout*).

Abaixo, estão listados alguns termos em inglês, que fazem parte da terminologia CVS, e seu significado:

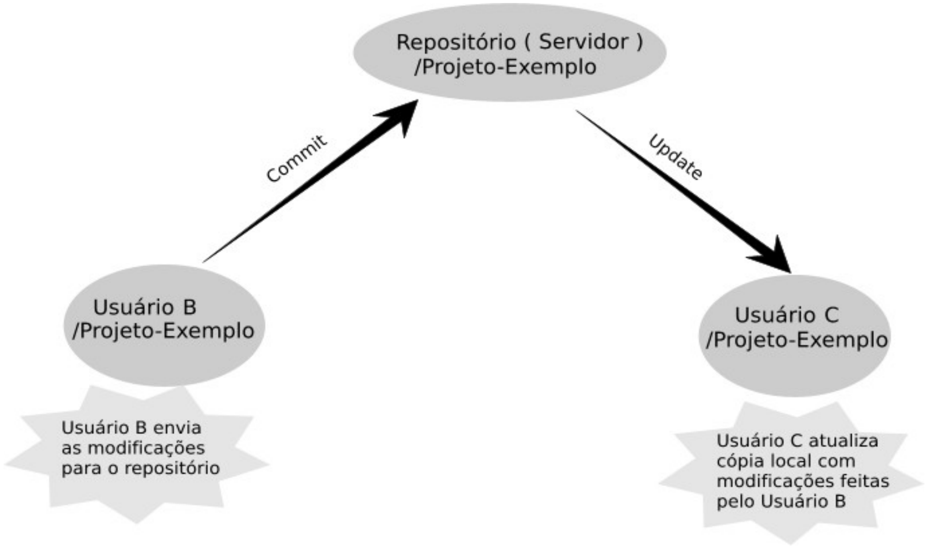
- ***Import***: geralmente é usado para designar a criação de um módulo inteiro dentro de um repositório CVS através do *upload* de uma estrutura de diretórios.
- ***Export***: é o *download* de um módulo inteiro a partir de um repositório CVS, sem os arquivos administrativos CVS. Módulos exportados, não ficam sob controle do CVS.



Import do Projeto para o Repositório no Servidor

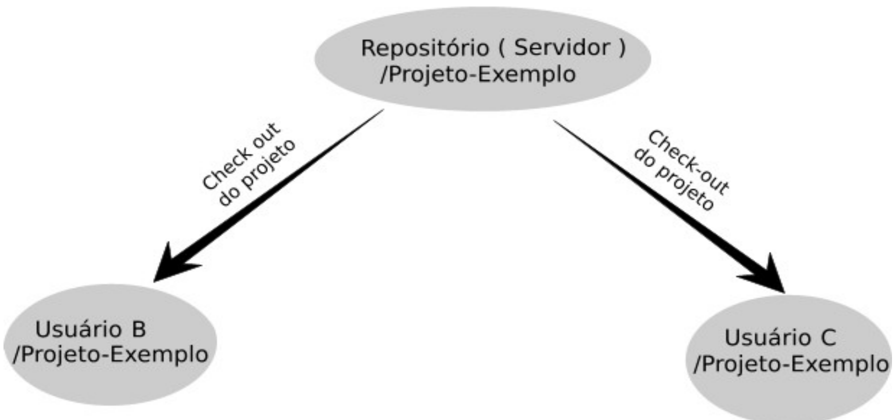
- ***Commit***: envio das modificações feitas pelo usuário ao repositório CVS.

- ***Update***: atualização da cópia local do trabalho através do download das modificações feitas por outros usuários no repositório.



Commit e Update de projeto

- ***Checkout***: normalmente é usado para denominar o primeiro *download* de um módulo inteiro a partir do repositório CVS.



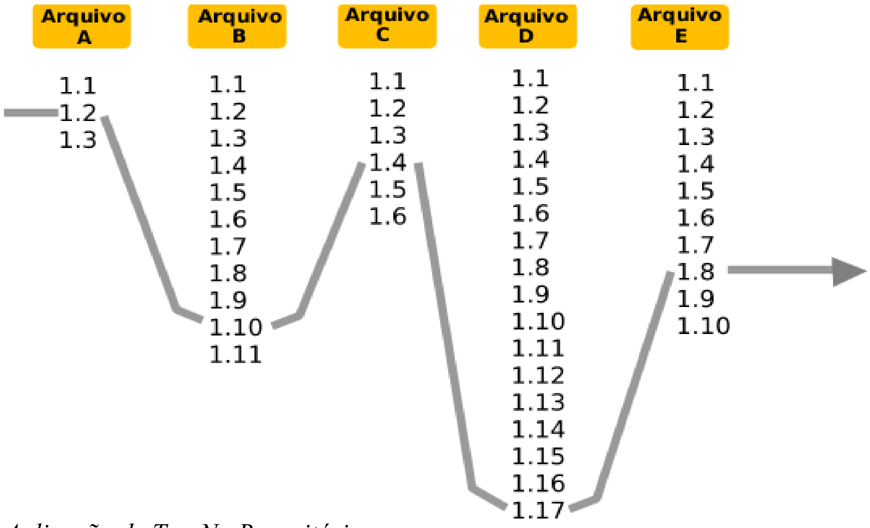
Checkout de Repositório a partir do Repositório no Servidor

- ***Module***: é uma hierarquia de diretórios. Geralmente um projeto de software existe como um simples módulo dentro do repositório.

- ***Release***: é a versão de um produto inteiro.

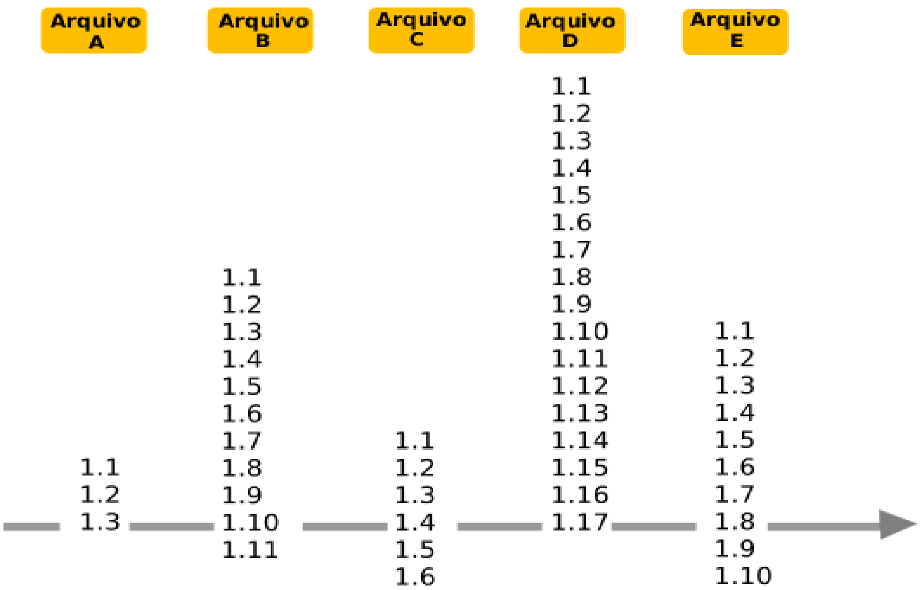
- ***Revision***: é a numeração atribuída pelo CVS a cada modificação de um arquivo.
- ***Tag***: é um nome simbólico dado para um conjunto de arquivos em um instante específico durante o desenvolvimento.

Geralmente uma tag é criada pelo projetista utilizando uma convenção para nomear as tags, a qual, poderá conter o nome do programa e ou o número da release. Então, por exemplo, o cvs 1.9 poderia ser nomeado como cvs1-9, cvs-1-9, cvs1_9, enfim, dependerá da convenção adotada no projeto. A figura abaixo ilustra o efeito de uma tag no repositório.



Aplicação de Tag No Repositório

Observando em uma linha horizontal a aplicação de uma tag, tem-se o seguinte resultado:

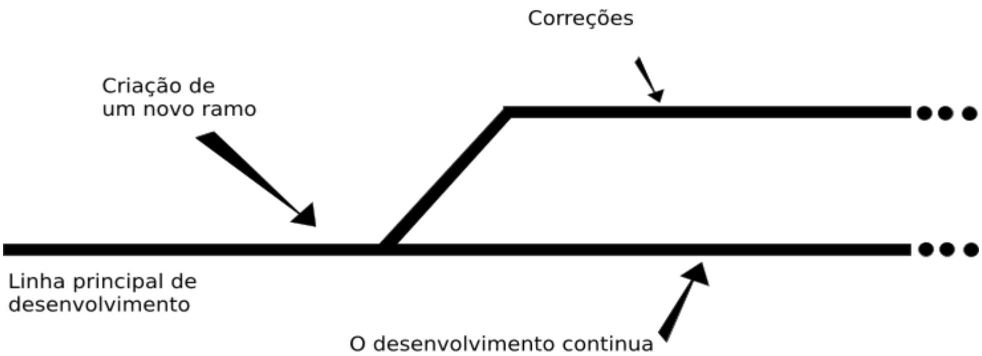


Tag aplicada ao repositório, Vista como uma linha reta

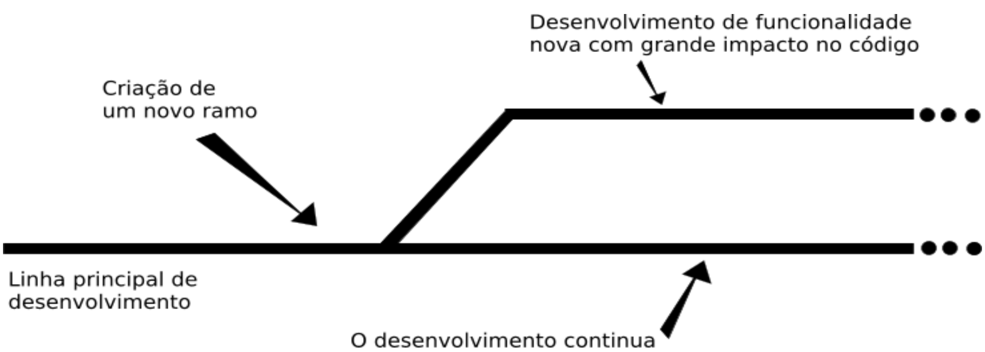
Ou seja, a tag aplicada refere-se aos arquivos: arquivo A versão 1.3, arquivo B versão 1.10, arquivo C versão 1.4, arquivo D versão 1.17. e arquivo E versão 1.8.

Dessa maneira obtém-se uma “foto” do repositório em determinado momento do desenvolvimento do produto.

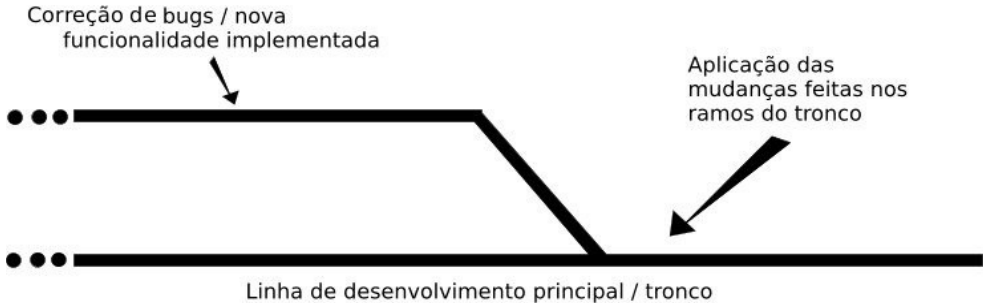
- ***Branch***: é uma ramificação no desenvolvimento, usada para descrever o processo de divisão dos arquivos de um projeto, em linhas de desenvolvimento independentes. Pode servir para teste de uma nova funcionalidade ou para projetos destinados a um cliente específico. É também chamado de ramo. A figura abaixo ilustra a criação de um novo branch do desenvolvimento, utilizado para aplicar correções de código, que a linha principal de desenvolvimento atual não comporta. Mais tarde essas correções aplicadas no branch criado, serão incorporadas à linha de desenvolvimento principal.



Uma outra fase onde a criação de um branch se faz necessária, é quando é preciso desenvolver uma nova funcionalidade que implica em um grande impacto no código atual. Sendo assim, cria-se um branch de desenvolvimento para essa funcionalidade nova e após finalizado o desenvolvimento dessa funcionalidade, incorpora-se o código do branch na linha de desenvolvimento principal. A figura abaixo indica o momento da criação do branch.



- ***Merge***: é a fusão de modificações feitas por diferentes usuários na cópia local de um mesmo arquivo. Sempre que alguém altera o código, é necessário realizar um *update* antes do *commit*, de modo que seja feito o *merge* - ou a fusão - das mudanças.

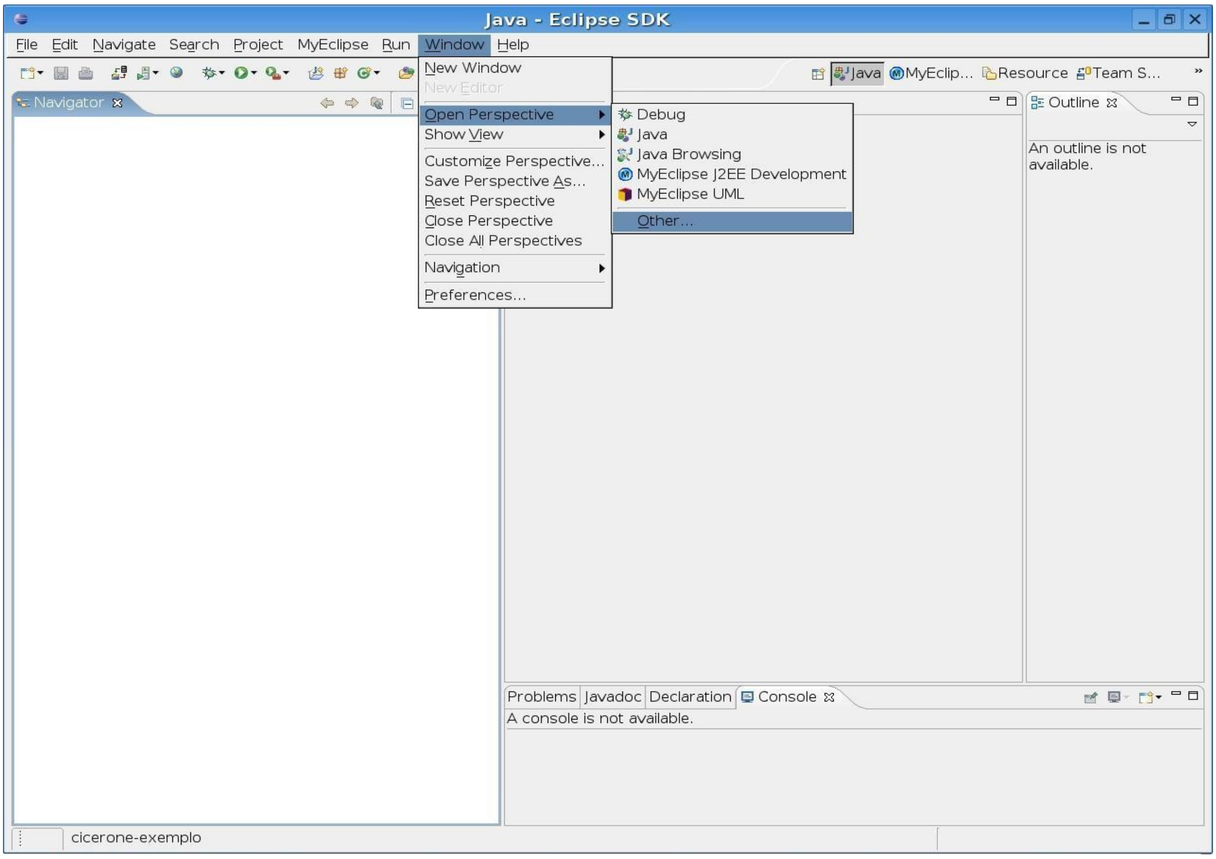


2 UTILIZAÇÃO DO CVS NO ECLIPSE

Para poder utilizar o CVS no Eclipse, algumas configurações são necessárias. Este documento tem como objetivo o passo a passo da configuração do CVS, dentro do Eclipse, e a navegação de arquivos baixados dele.

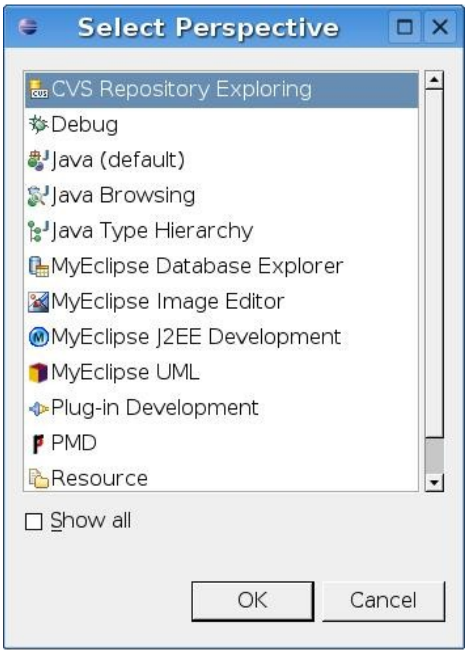
2.1 Configurando o repositório

O Eclipse possui diversas perspectivas de trabalho. Para configurar o usuário do CVS é preciso ir na perspectiva do CVS conforme a figura *Abrindo Perspectiva CVS*:



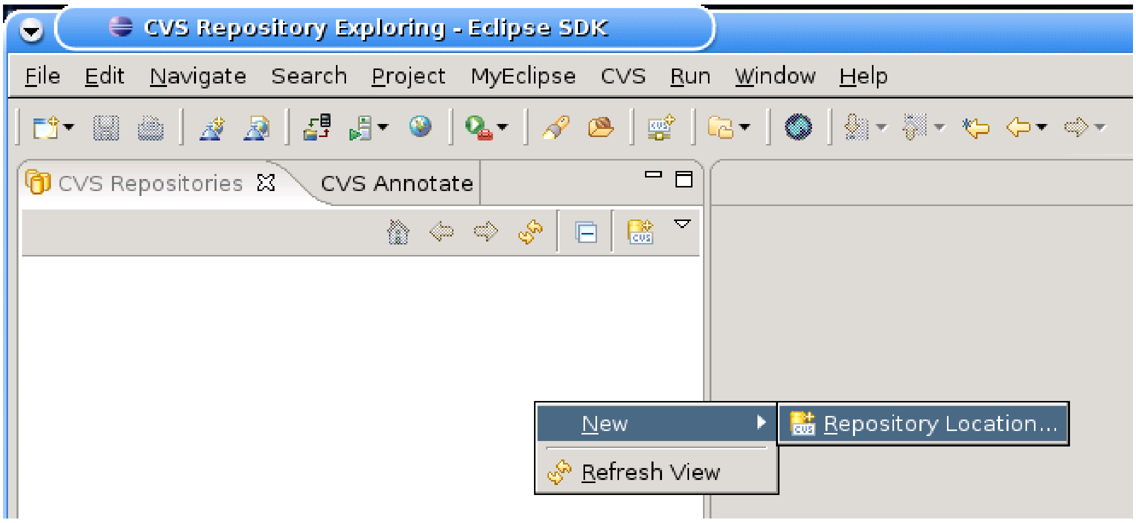
Abrindo Perspectiva CVS

Escolher a opção *CVS Repository Exploring* conforme ilustra a figura *Selecionando Perspectiva*:



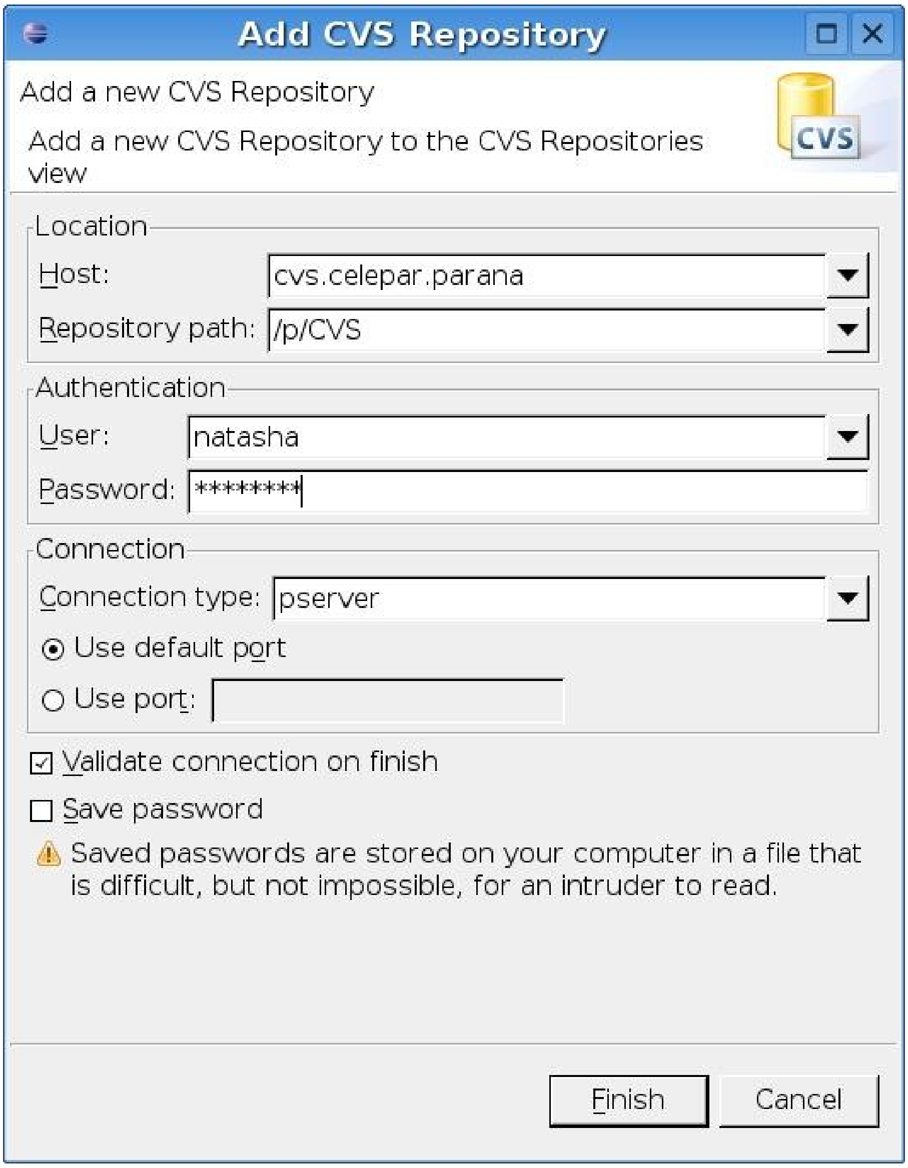
Selecionando Perspectiva

Automaticamente o Eclipse carregará a perspectiva do CVS após o clique no botão *OK*. Dentro dessa perspectiva deve-se clicar com o botão direito dentro da aba *CVS Repositories* e escolher as opções *New ->Repository Location...* conforme indica a figura *Novo Repositório*:



Novo Repositório

Após isso, aparecerá a tela *Add CVS Repository*, onde devem ser configurados: o endereço do servidor (*cvs.celepar.parana*), o diretório do cvs (*/p/CVS*), o usuário e a senha conforme exemplifica a figura *Configuração de Repositório*:



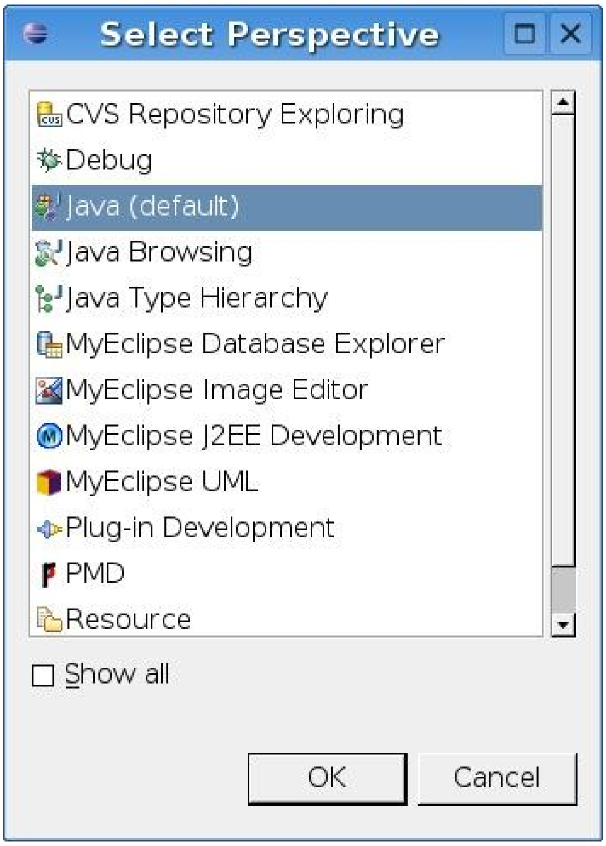
Configuração do Repositório

Após preenchimento dos campos clicar no botão *Finish* para concluir a configuração do CVS.

2.2 Acessando um novo Projeto

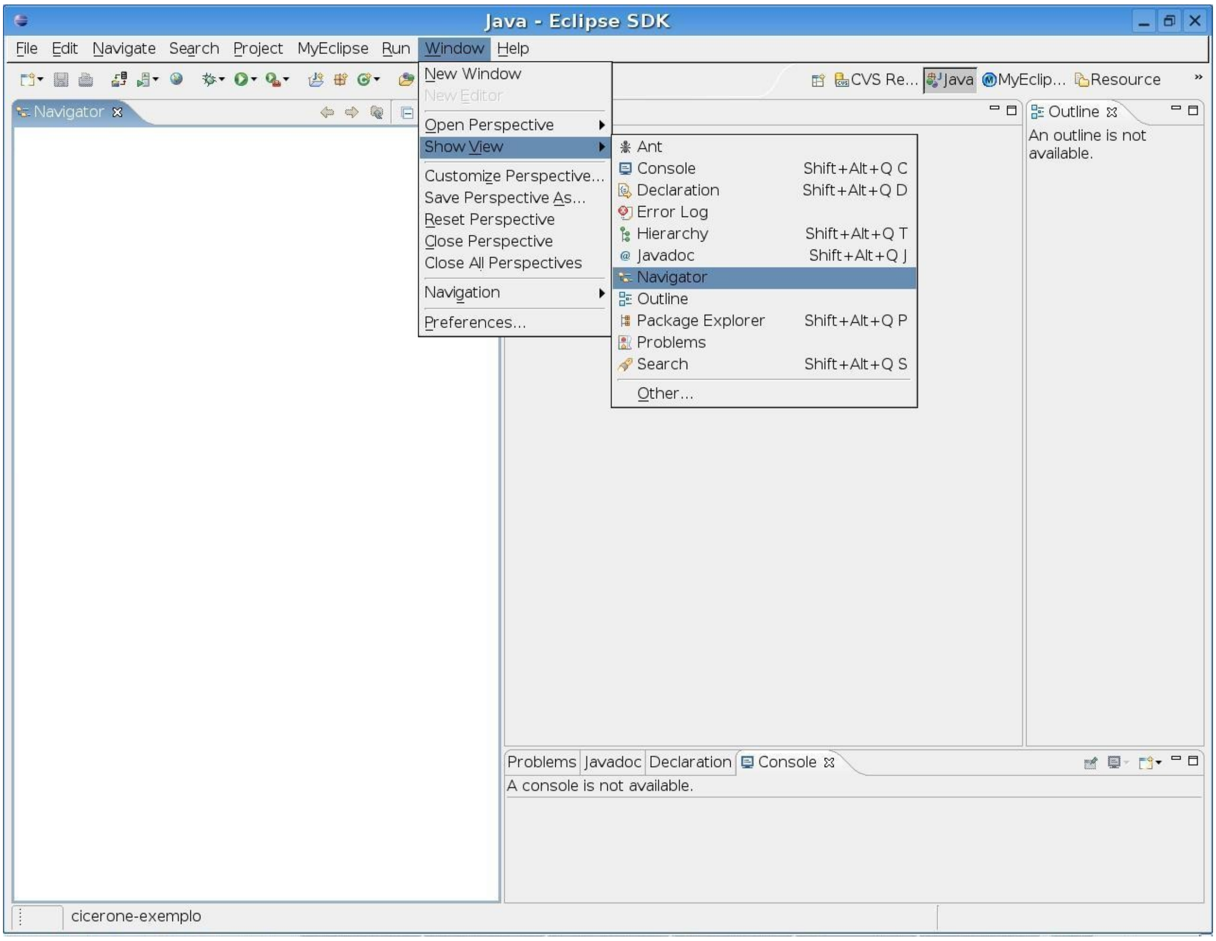
Como explicado anteriormente, o Eclipse tem diversas perspectivas e a escolhida para

visualização de projetos é a Java. Para mudar para essa perspectiva é preciso escolher a opção *Java* (default) conforme ilustra a figura *Seleção de Perspectiva Java*:



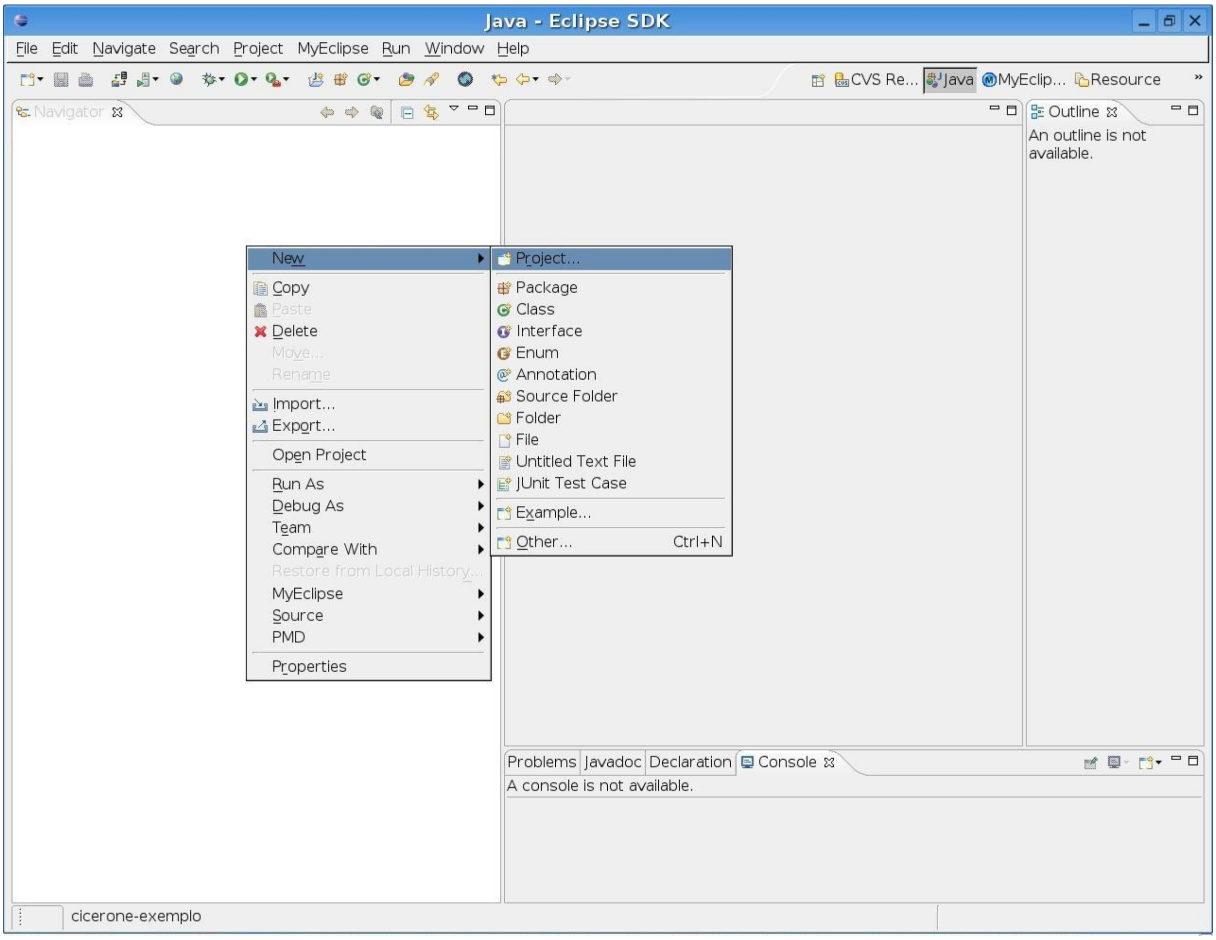
Seleção de Perspectiva Java

Para uma melhor visualização dos projetos, é necessário abrir a visão de navegação. Ela se encontra na opção *Window* -> *ShowView* -> *Navigator* de acordo com a figura *Visão Navegador*:



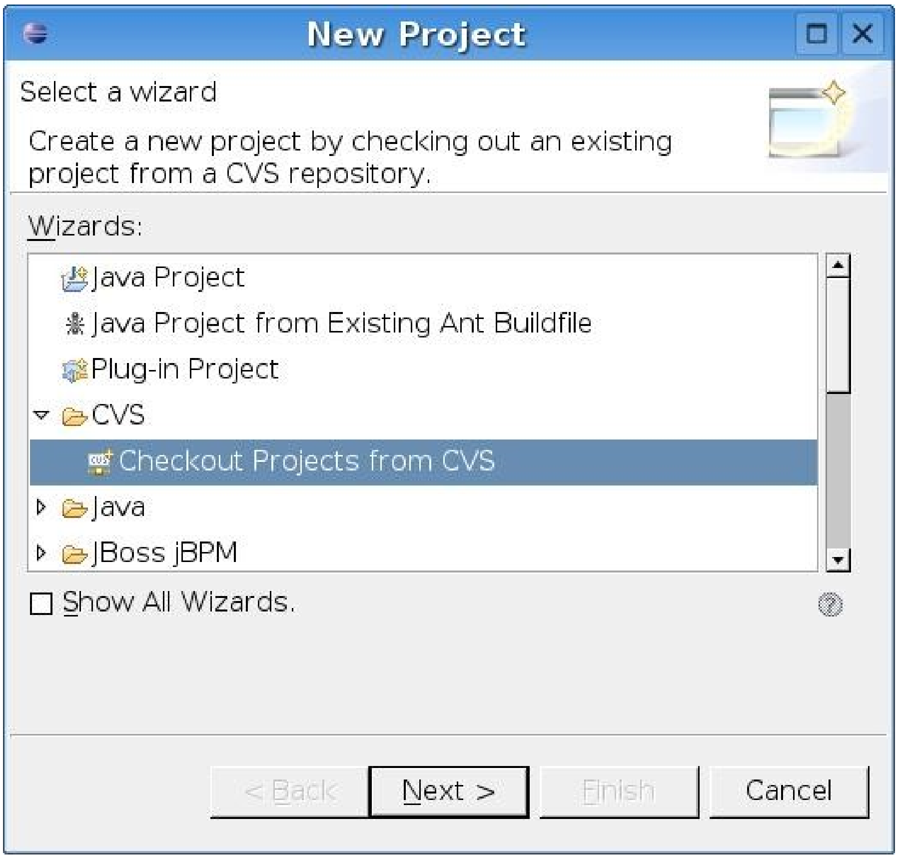
Visão Navegador

Uma nova aba *Navigator* irá aparecer, clique com o botão direito e em *New -> Project* conforme mostra a figura *Novo Projeto*:



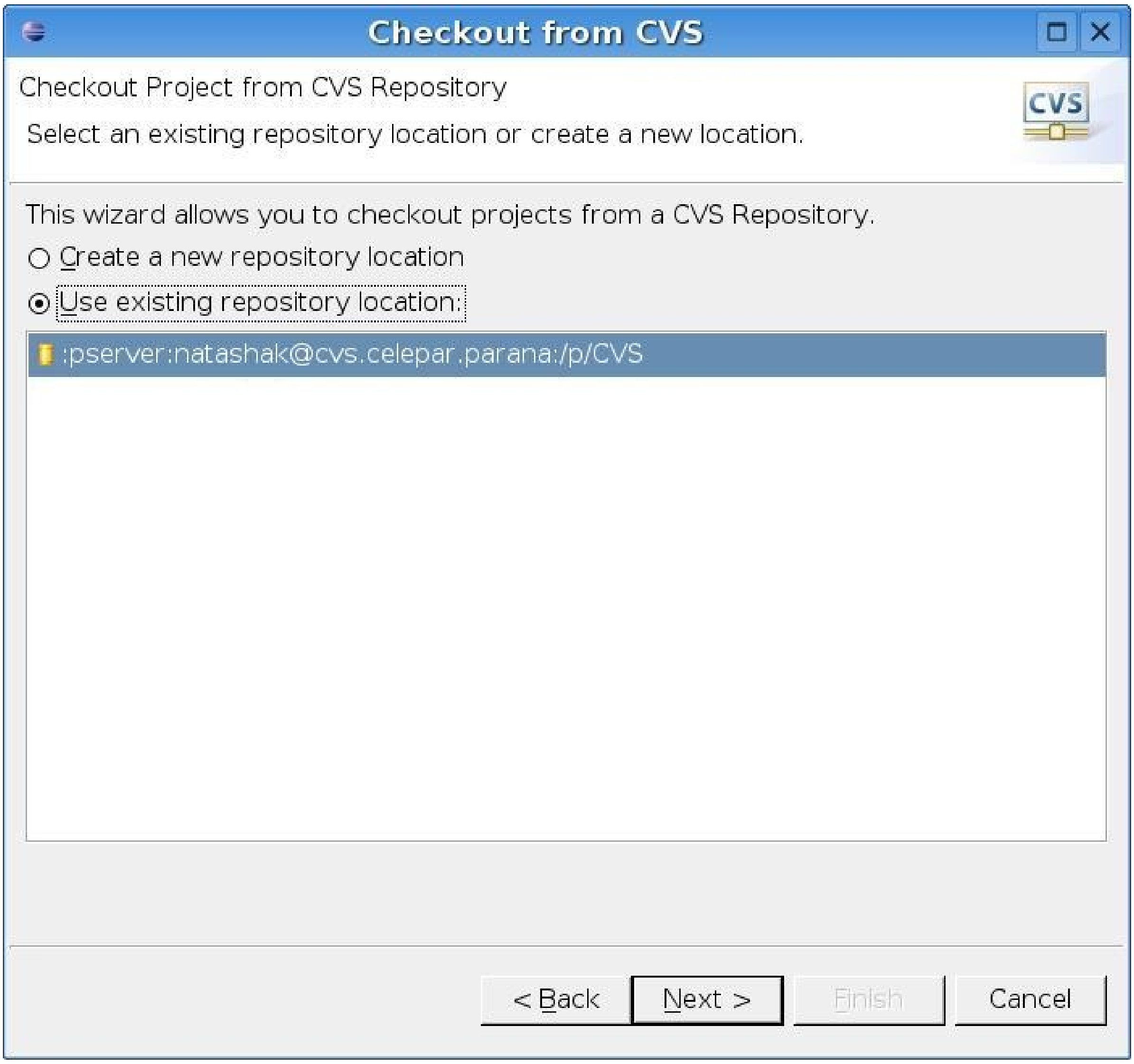
Novo Projeto

A tela da figura *Checkout do CVS* irá aparecer. Nela é preciso escolher a opção "Checkout Projects from CVS" e clicar em "Next".



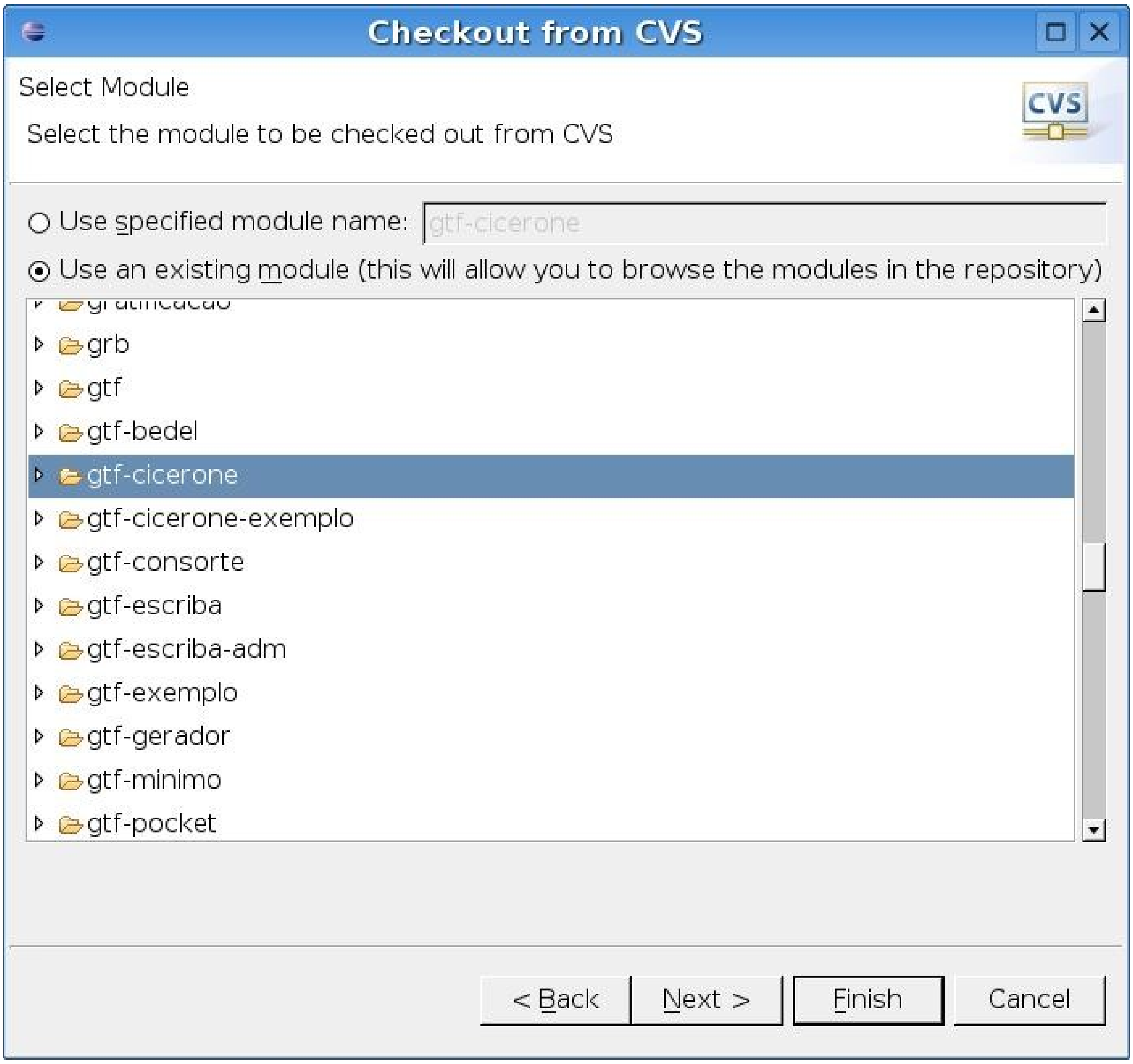
Checkout do CVS

Na tela seguinte, ilustrada na figura *Escolhendo Repositório*, mostra uma lista de repositórios (usuários configurados naquele Eclipse para acessar o CVS) e deve-se escolher o usuário para qual, o analista responsável pelo projeto, solicitou a permissão de acesso.



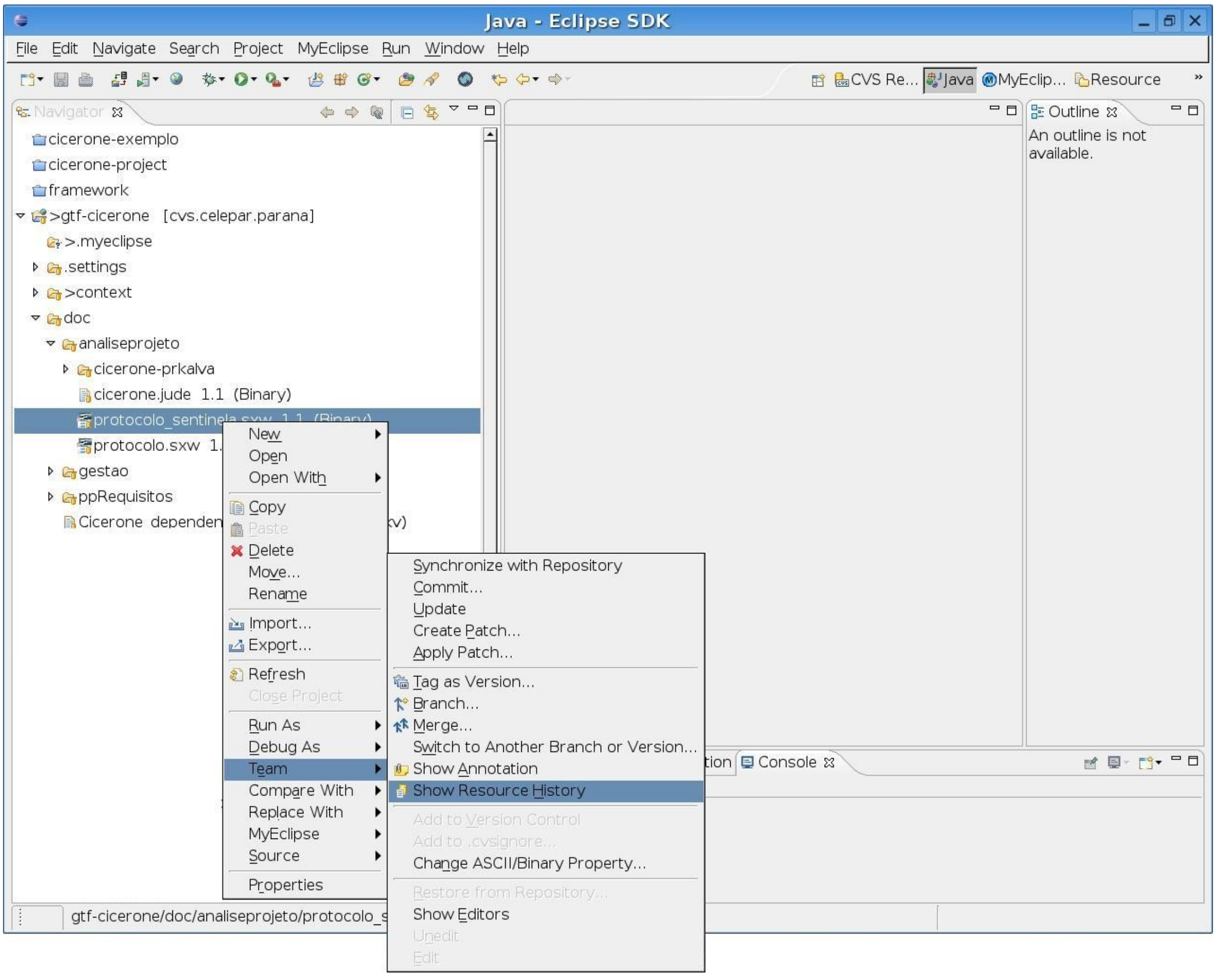
Escolhendo Repositório

Após clique no botão *Next* irá aparecer uma tela, ilustrada na figura *Escolhendo Projeto*, que lista todos os projetos contidos no CVS. Deve-se escolher o projeto que o analista requisitou autorização e clicar em “Finish”.



Escolhendo Projeto

Será criada na aba *Navigator* uma pasta com o nome do projeto. Dentro dela encontram-se todos os arquivos referentes a esse projeto. Para saber quando um determinado arquivo foi atualizado pela última vez, basta clicar com o botão direito em cima do arquivo e ir em *Team->Show Resource History* de acordo com a figura *Mostrando o Histórico*:

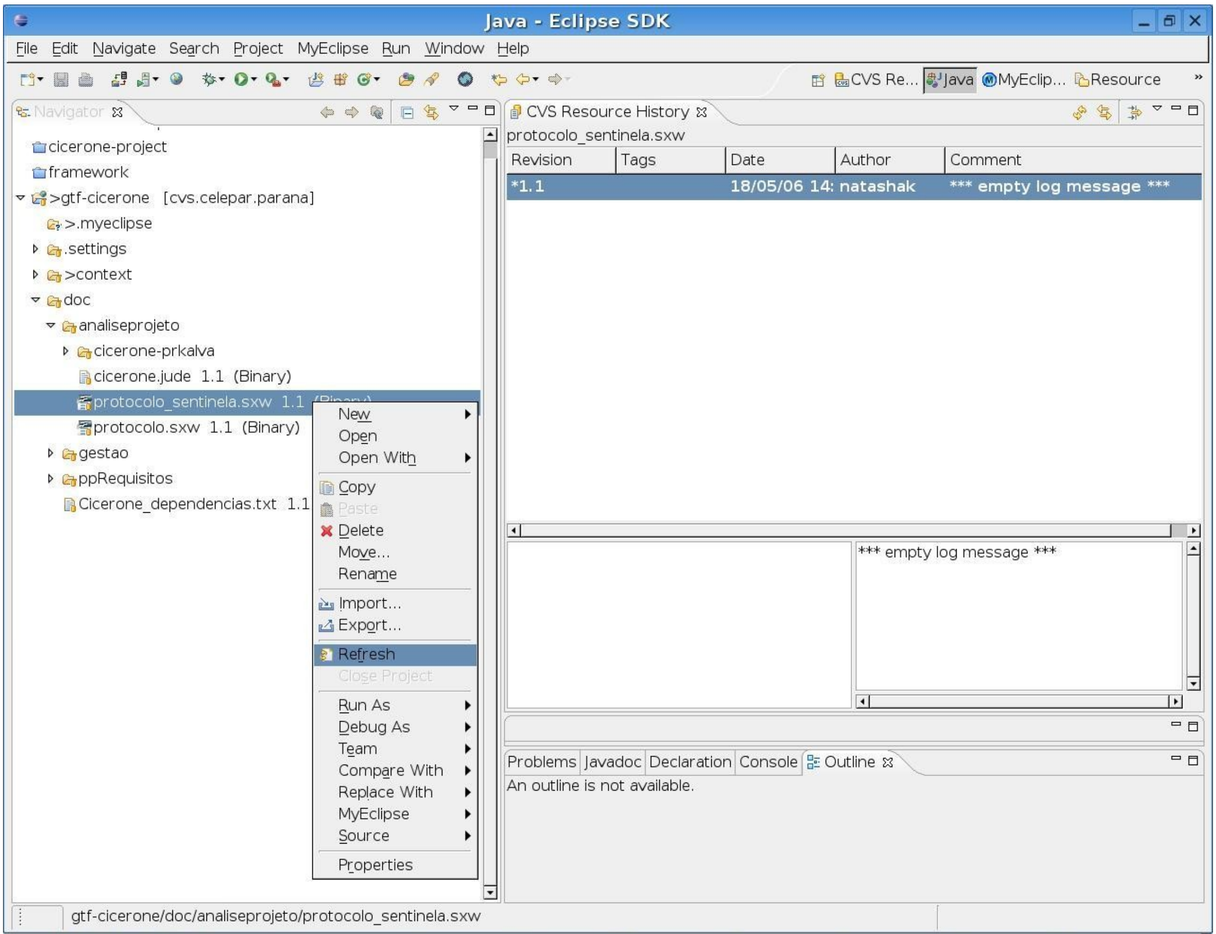


Mostrando Histórico

2.3 Enviando arquivos para o CVS

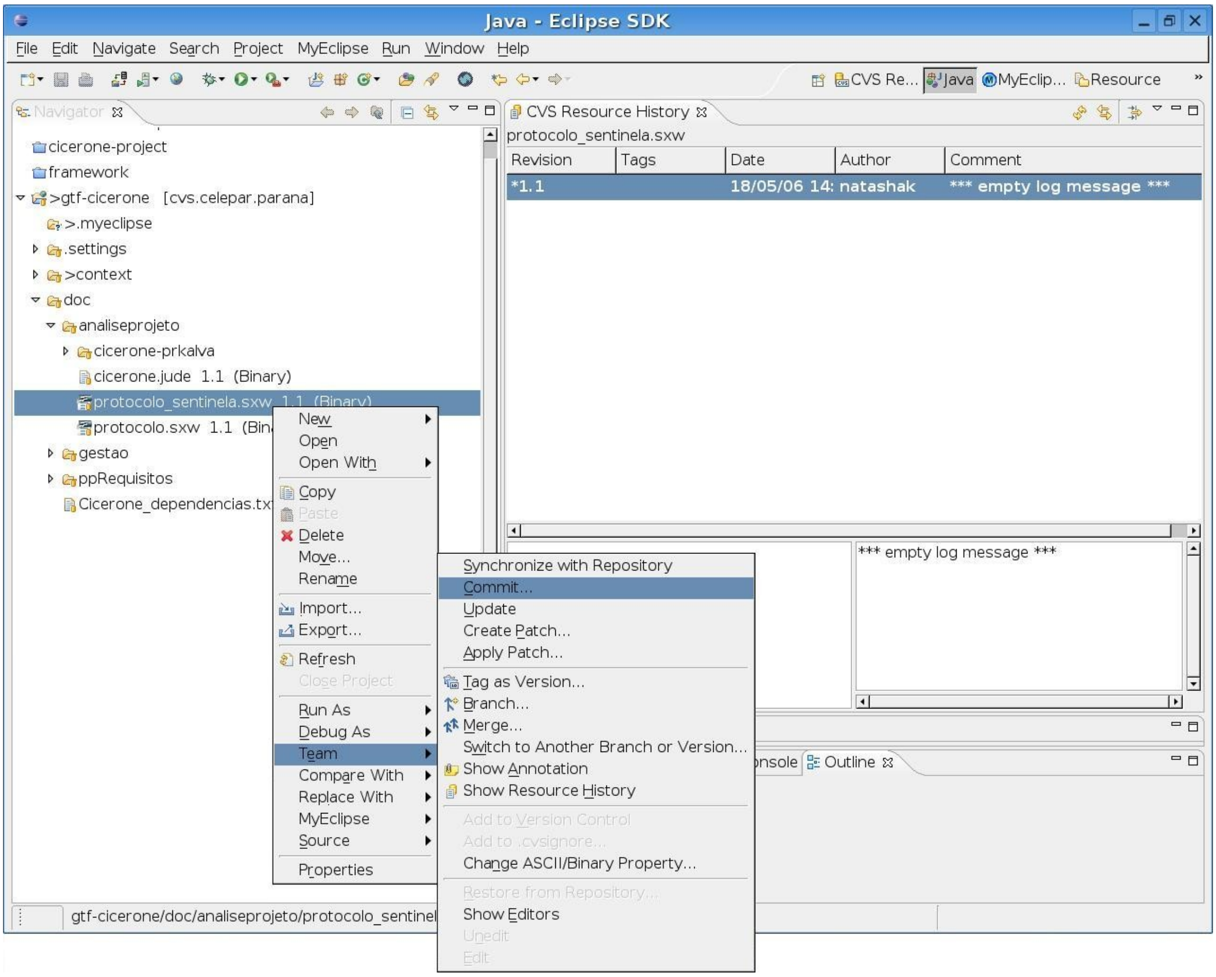
O procedimento de envio de arquivos para o CVS depende do usuário ter permissão de

gravação no CVS. Antes de enviar o arquivo é preciso atualizar o seus arquivos com o Eclipse. Para fazer isso é preciso clicar com o botão direito no arquivo e em *Refresh* conforme mostra a figura *Atualizando Área de Trabalho*:



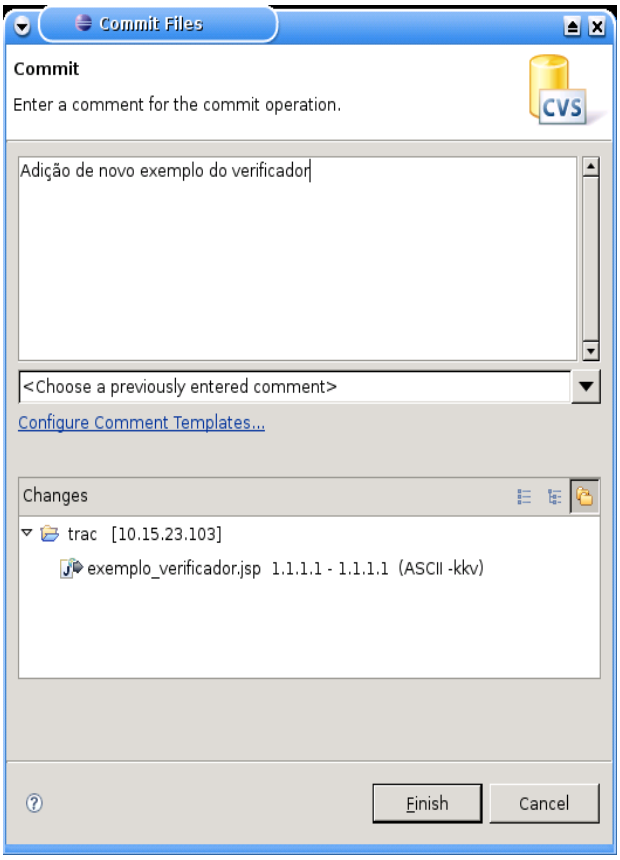
Atualizando Área de Trabalho

Para enviar o arquivo para o CVS é preciso clicar com o botão direito em cima do arquivo *Team->Commit* conforme a figura *Commit de Arquivo*:



Commit de Arquivo

No próximo passo surgirá a janela encontrada na figura Mensagem de Commit.

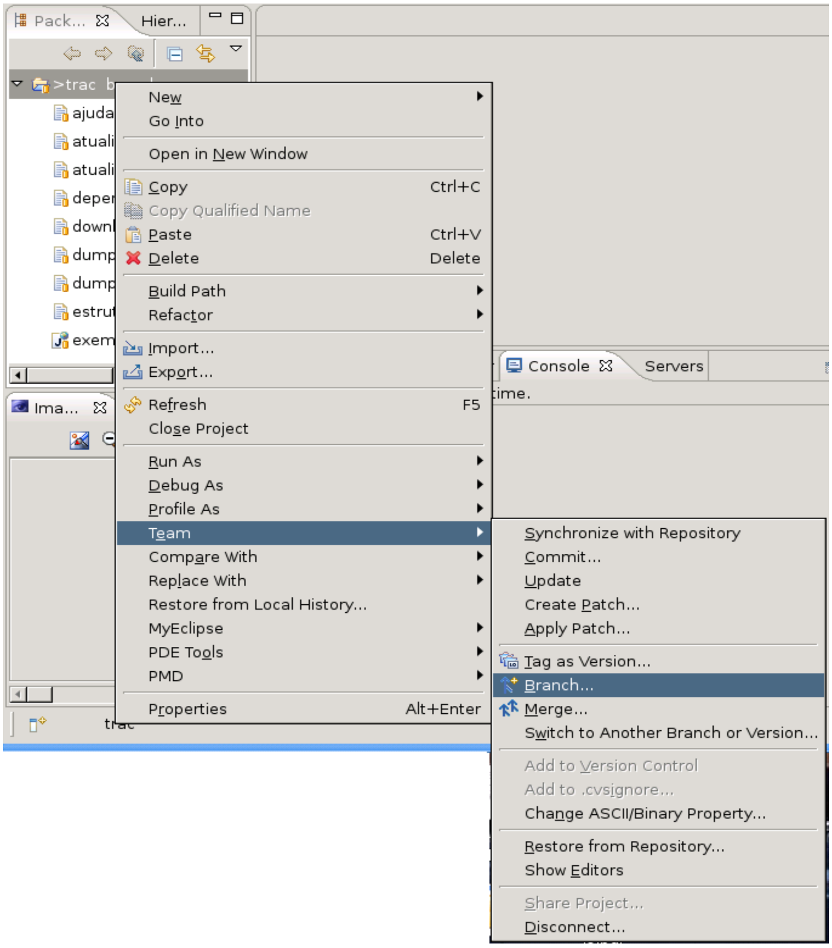


Mensagem de Commit

Observe que a primeira caixa de texto que aparece é utilizada para colocar uma mensagem que indica quais as modificações executadas neste commit. A caixa abaixo (*Changes*) indica quais foram os arquivos modificados.

2.4 Criando um Branch/Ramo

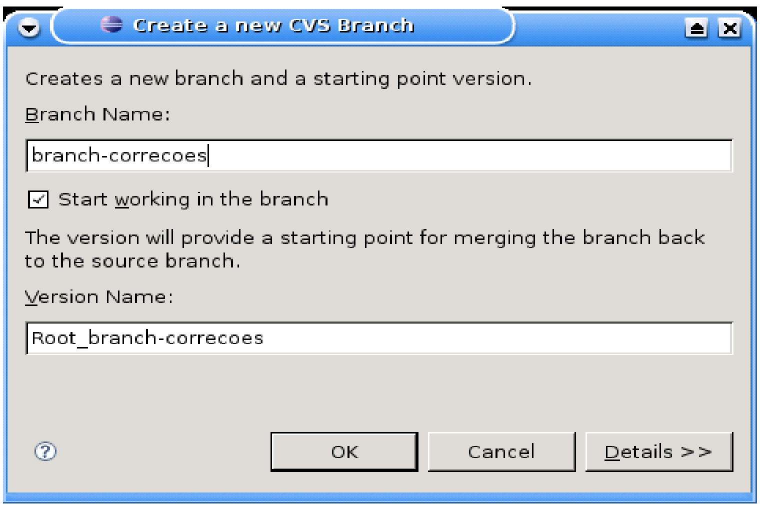
A criação de um branch utilizando o eclipse, seguem alguns passos bem simples. O primeiro



Criando um Branch

deles é, tendo o projeto no workspace, clicar com o *botão direito* do mouse sobre o nome do projeto e seguir a ordem: *Team* -> *Branch.*

A seguir, surgirá uma janela como a ilustrada abaixo:



Onde deverá ser informado o nome do branch. O eclipse automaticamente cria uma *tag* para identificar o ponto onde foi criado o branch. Note que o *check-box* “Start working in the branch” está selecionado por padrão. Dessa maneira, ao criar o branch começa-se a trabalhar nele.

A figura *Trabalhando no Branch* ilustra o branch recém criado.



Trabalhando no Branch

Comparando com a linha principal de desenvolvimento:

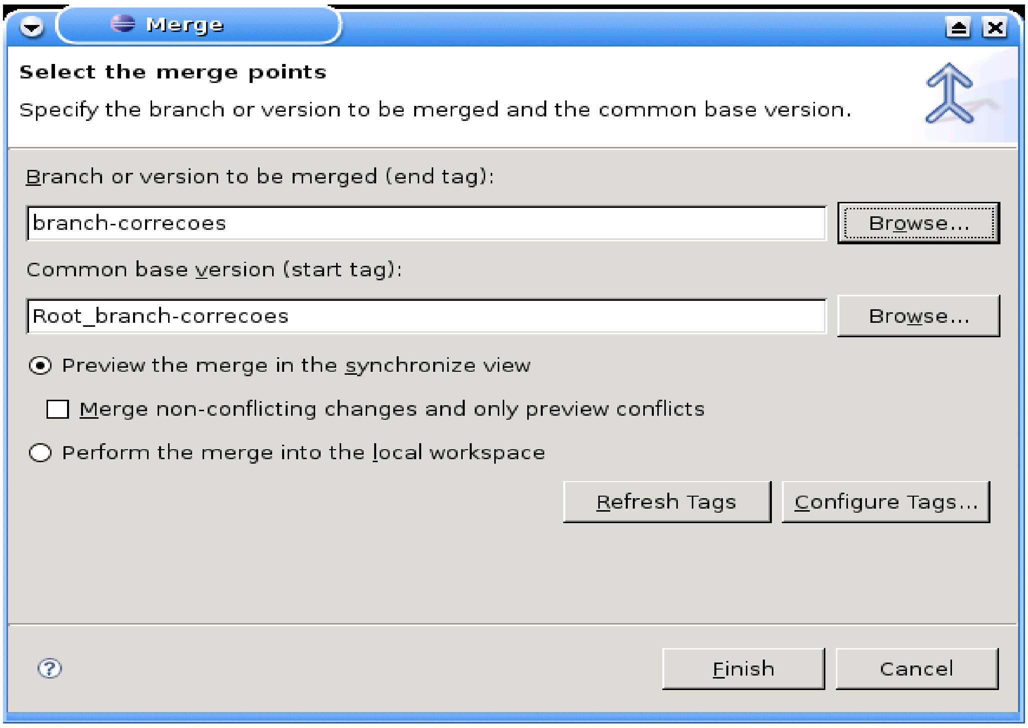


Trabalhando no HEAD

Observe que ao lado do nome do projeto o eclipse mostra o nome do ramo (*branch-correcoes)* ou o nome do servidor *([10.15.23.103])* caso esteja na linha principal de desenvolvimento (*HEAD*).

Após efetuar as modificações no branch, normalmente deseja-se fazer o *merge* ou agrupamento da funcionalidade/correção de bugs no projeto principal.

Para isso, deve-se clicar com o *botão direito* do mouse sobre o título do projeto e seguir os menus: *Team* -> *Merge*. Surgirá uma janela como indica a figura *Merge do Branch*:

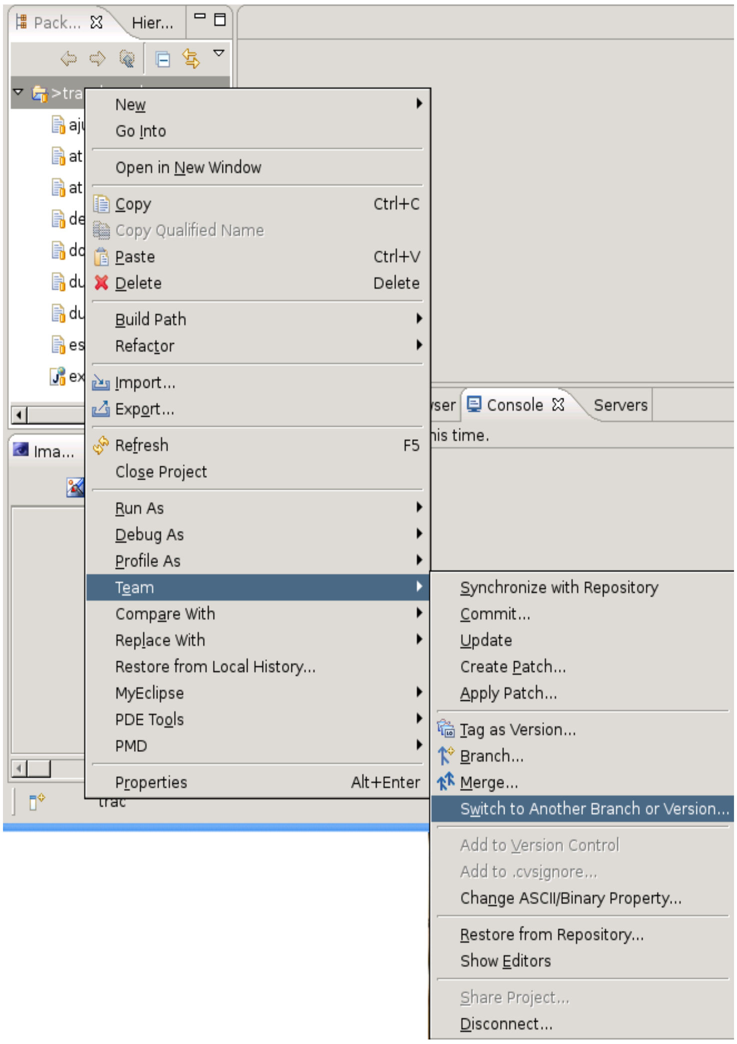


Merge do Branch

Escolha o branch clicando no botão *browse* e a seguir no botão *finish.*

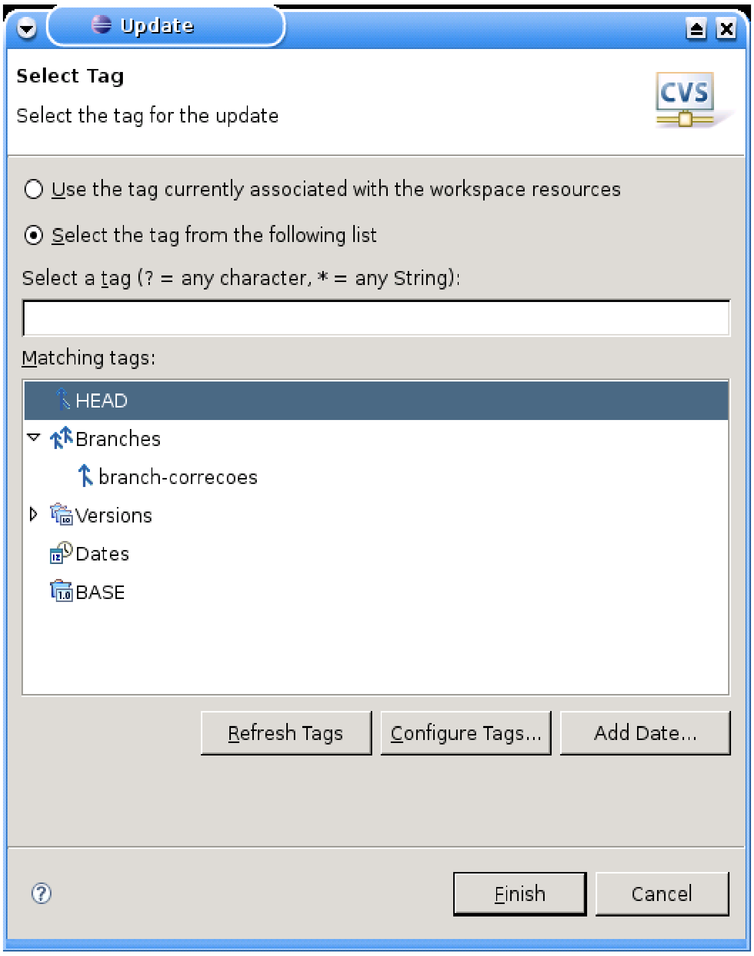
Pode ser necessário entretanto, trocar entre o *branch* criado e a linha de desenvolvimento principal (HEAD) do projeto.

Para isso, basta clicar com o *botão direito* do mouse sobre o título do projeto e clicar em: *Team -> Switch to Another Branch or Version* como ilustrado na figura *Trocando de Branch*.



Trocando de Branch

A seguir, selecionar o branch que se deseja tornar atual, ou ainda HEAD, caso desejar ir para a linha de desenvolvimento principal.



Seleção do Projeto

Ao clicar no botão *Finish,* o projeto estará de volta à linha de desenvolvimento principal (*HEAD*)